

September 2003

Divide et Impera - Ein Peer-to-Peer basiertes Geschäftsmodell für Streaming Media

Nick Gehrke

Universität Göttingen, ngehrke@uni-goettingen.de

Markus Burghardt

Universität Göttingen

Matthias Schumann

Universität Göttingen

Follow this and additional works at: <http://aisel.aisnet.org/wi2003>

Recommended Citation

Gehrke, Nick; Burghardt, Markus; and Schumann, Matthias, "Divide et Impera - Ein Peer-to-Peer basiertes Geschäftsmodell für Streaming Media" (2003). *Wirtschaftsinformatik Proceedings 2003*. 25.

<http://aisel.aisnet.org/wi2003/25>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2003 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

In: Uhr, Wolfgang, Esswein, Werner & Schoop, Eric (Hg.) 2003. *Wirtschaftsinformatik 2003: Medien - Märkte - Mobilität*, 2 Bde. Heidelberg: Physica-Verlag

ISBN: 3-7908-0111-9 (Band 1)

ISBN: 3-7908-0116-X (Band 2)

© Physica-Verlag Heidelberg 2003

Divide et Impera – Ein Peer-to-Peer basiertes Geschäftsmodell für Streaming Media

Nick Gehrke, Markus Burghardt, Matthias Schumann

Universität Göttingen

Zusammenfassung: Peer-to-Peer (p2p) Systeme nutzen dezentrale Ressourcen aus, um bestimmte Services innerhalb eines p2p Netzes zu erbringen. In dem vorliegenden Beitrag wird ein p2p basiertes Geschäftsmodell vorgestellt, welches die kostenpflichtige asynchrone Auslieferung von Streams ermöglicht, ohne eine zentrale contentliefernde Einheit zu benötigen. Dabei wird die Entlohnung des Rechteinhabers des Streams berücksichtigt. Peers, die Teile eines Streams an andere „Konsumentenpeers“ ausliefern, werden darüber hinaus ebenfalls entlohnt, da diese Peers erstens aus betriebswirtschaftlicher Sicht Produktionsfaktoren darstellen und zweitens ein Anreiz zum Bereitstellen von Speicherplatz und Netzwerkbandbreite aus Sicht eines „Anbieterpeers“ notwendig ist. Der Beitrag stellt den konzeptionellen Aufbau eines solchen p2p Systems dar. Anschließend wird eine Analyse zur Abschätzung der benötigten dezentralen Ressourcen (vor allem der Bandbreite) durchgeführt, damit genügend kritische Masse an Produktionsfaktoren für ein sinnvolles Betreiben des p2p Systems vorhanden ist. Eine ökonomische Analyse schließt den Beitrag ab.

Schlüsselworte: Peer-to-Peer, Streaming Media, Geschäftsmodell

1 Einleitung und Gang der Untersuchung

Das Peer-to-Peer Paradigma ist in den letzten Jahren sehr populär geworden. Hauptsächlich das erste weitverbreitete Filesharing-System Napster [Shir01 S. 21ff] verursachte aufgrund des kostenlosen Musikbezugs den ersten „Peer-to-Peer-Hype“. Nach Napsters Einstellung aufgrund einer gerichtlichen Entscheidung entstanden viele weitere – und sehr viele weiterentwickelte – Filesharing-Systeme. Aufgrund des durch diese Systeme leicht möglich gewordene Raubkopieren wird das Thema „Peer-to-Peer“ oftmals im Zusammenhang mit Verletzungen des Urheberrechts in Verbindung gebracht. Peer-to-Peer Systeme können jedoch auch für andere Zwecke verwendet werden. So finden sich in der Literatur beispielsweise Ansätze zur Konstruktion elektronischer Marktplätze mit Hilfe von Peer-to-Peer Architekturen [Ge+02]. Aber auch Themen wie Grid Computing [AnKu02, Bark01 S. 269] und Instant Messaging [Leu02, S. 139ff] sind eng mit Peer-to-Peer Systemen verbunden.

In diesem Beitrag wird eine weitere Anwendung – ein p2p-Netz zum Ausliefern von Streaming-Media - herausgearbeitet. Das Streaming von multimedialen Inhalten [Zwi01, S. 164] stellt besondere Anforderungen an Bandbreite und Verbindungsqualität, da im Gegensatz zu herkömmlichen Datenübertragungen spezielle Zeitrestriktionen einzuhalten sind. Zielsetzung des vorgestellten Geschäftsmodells ist es, Streams nicht durch zentrale Server auszuliefern, sondern durch dezentrale Peers, die ihre freie Bandbreite zur Auslieferung von Streams zur Verfügung stellen. Wesentliche Motivation ist dabei, dass Streaming enorme Anforderungen an Hardware stellt und außerdem sehr schnell die Kapazitäten selbst mächtiger zentraler Server erschöpft. Ziel ist daher die Vermeidung großer Infrastrukturkosten durch Nutzung brachliegender, dezentraler Ressourcen.

Nachfolgend wird zwischen zwei Arten von Streaming unterschieden. Einerseits kann ein Stream mehrfach gleichzeitig an der selben Stelle an verschiedene Empfänger übertragen werden. Der Stream wird also synchron an viele Konsumenten übermittelt. Diese Art des Streaming eignet sich beispielsweise für Live-Übertragung oder Radiosendungen. Auf der anderen Seite existiert das asynchrone Streaming. Hierbei kann ein Konsument den Stream zu einer beliebigen Zeit anfordern und anschließend empfangen (Streaming on demand). Diese Vorgehensweise eignet sich überwiegend für den Empfang von Filmen oder von Musik, die ein Konsument spontan anfordert. Das folgende Geschäftsmodell fokussiert dabei das asynchrone Streaming und ermöglicht das spontane Empfangen von (kostenpflichtigen) Streams. Der vorliegende Beitrag geht dabei nicht auf die technische Umsetzung des benötigten Peer-to-Peer Systems ein, sondern bleibt auf der konzeptionellen Ebene. Insbesondere werden nacheinander folgende Aspekte beleuchtet:

- *Beschreibung des Geschäftsmodells.* Die Umschreibung des Geschäftsmodells beinhaltet den konzeptionellen Aufbau des p2p-Systems und die Erläuterung des Erlösmodells. Die beteiligten Organisationseinheiten werden dargestellt und der Ablauf einer „Streaming-Session“ beschrieben. Das Erlösmodell klärt, welche am Wertschöpfungsprozess beteiligten Organisationseinheiten wie entlohnt werden. Das vorgestellte Erlösmodell konzentriert sich dabei insbesondere auf die Entlohnung etwaiger Rechteinhaber und beschäftigt sich auch damit, wie den dezentralen Peers Anreize geboten werden können, um die Produktionsfaktoren „Speicherplatz“ und „Bandbreite“ in ausreichendem Maße vorhalten zu können.
- *Ressourcenanalyse.* Die Ressourcen oder Produktionsfaktoren stellen die dezentralen Peers zur Verfügung. Ein Stream wird in der Regel nicht nur von einem Konsumenten zur Zeit empfangen. Die Regel wird sein, dass ein Stream gleichzeitig von mehreren Konsumenten, aber an verschiedenen Stellen, empfangen wird. Hierfür muss insgesamt genügend Bandbreite zur Verfügung stehen. Insofern ist es wichtig zu wissen, wie viele Peers benötigt werden und wie ein Stream auf die vielen Peers verteilt werden muss. Diese Ressourcenanalyse gibt Aufschluss über die notwendige kritische Masse des p2p Systems.

- *Ökonomische Bewertung des Geschäftsmodells.* Nach der Ressourcenanalyse wird das vorgestellte Geschäftsmodell ökonomisch bewertet. Die wesentlichen ökonomischen Charakteristika des Systems werden herausgearbeitet und Kritikpunkte aufgedeckt.

2 Technologien für Streaming Media

2.1 Streaming Media Grundlagen

Bevor auf das p2p Geschäftsmodell eingegangen wird, sollen einige technische Grundlagen zum Thema „Streaming Media“ diskutiert werden. Streaming Media Technologien stellen höhere Anforderungen an Netze und Ressourcen als Datenflüsse ohne zeitliche Restriktionen. Diese Anforderungen werden auch Quality of Service (QoS) genannt und sind im Einzelnen [Cou+02, S. 714]:

- *Bandbreite.* Die Bandbreite gibt die Geschwindigkeit an, mit der ein Multimedia-Stream transportiert wird. Multimedia-Streams verbrauchen hohe Bandbreiten. Insofern sollte die Bandbreite während der Übertragung erstens hinreichend hoch und zweitens konstant sein.
- *Latenz.* Als Latenz wird die Zeitspanne bezeichnet, die ein Stream vom Sender zum Empfänger benötigt (Transportlaufzeit). Die Latenz sollte für Streams möglichst klein sein und keinen großen Variationen unterliegen, damit die Streamelemente ohne Verzögerung eintreffen, sodass der Stream ohne Unterbrechung bzw. ohne Lücken konsumiert werden kann.
- *Verlustrate.* Die Verlustrate ist ein Ausdruck dafür, wie viele Streamelemente beim Empfänger nicht verarbeitungsbereit ankommen. Der Verlust beruht dabei in der Regel nicht auf Datenverlusten während des Transports z.B. durch Störungen, sondern vielmehr darauf, dass Streamelemente später beim Empfänger eintreffen als ihre geplante Position in der Abspielreihenfolge. Solche Streamelemente sind zwangsläufig unbrauchbar und werden beim Empfänger verworfen.

Für Größenordnungen von Bandbreite, Latenz und Verlustrate siehe [Cou+02, S. 712].

Audio- und besonders Videodateien erreichen bei akzeptabler Qualität schnell enorme Größenordnungen (z.B. benötigt ein MPEG-1-Stream ca. 1,5 Mbps). Um einen Stream in ausreichender Zeit über ein Netzwerk zu schicken, ist insofern eine vorangehende Komprimierung notwendig. Das jeweilige Komprimierungsverfahren hängt von der Art des Streams ab. Im Bereich von Audioinhalten haben sich Komprimierungsverfahren wie z.B. MPEG-Audio (Motion Picture Expert

Group), das AU-Format (m-law Format) oder das AIFFC Format (Audio Interchange File Format Compressed) verbreitet. Bei Videoinhalten ist z.B. AVI (Audio Video Interface), MPEG, DivX oder MOV (Apple QuickTime) zu nennen. Der Stream wird vor dem Ausliefern mit Hilfe eines so genannten Codecs komprimiert und dann über das Netzwerk verschickt. Beim Empfänger angekommen, wird der Stream mit einem Entschlüsselungs-Codec entkomprimiert und an ein entsprechendes Ausgabegerät weitergeleitet. Abbildung 1 illustriert den Ablauf des Streamings.

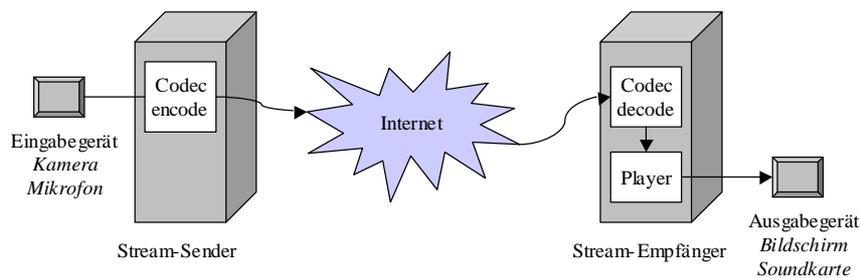


Abbildung 1: Der Prozess des Streamings

Für das Ausliefern von Streams in gebotener Dienstgüte sind spezielle Protokolle notwendig. Diese Protokolle setzen in der Regel über den Protokollen TCP/IP bzw. UDP/IP an und folgen insofern dem in der Informatik üblichen Schichtenmodell. Die im Zusammenhang mit Streams verwendeten Protokolle sind [Erpen99; Cou+02, S. 99; Stein99, S. 495ff]:

- Das RTP (Realtime Transport Protocol) baut für gewöhnlich auf UDP auf. Das RTP Protokoll übermittelt z.B. die Art des Streams (Audio/Video) und nummeriert einzelne Pakete durch. Auch Timing-Anforderungen für den Stream werden transportiert. RTP selbst kann jedoch keine rechtzeitige Zustellung der Pakete garantieren, so dass eine hinreichende QoS nicht gewährleistet werden kann.
- Das RTCP (Realtime Controll Protocol) steht im Zusammenhang mit dem RTP. Mit diesem Protokoll sind Rückmeldungen an den Sender möglich, so dass die Güte des Datenflusses an den Sender übertragen werden kann.
- Das RTSP (Realtime Streaming Protocol) bietet Funktionen zum notwendigen Caching des Streams und ermöglicht Load-Balancing. Somit wird das Beziehen eines Streams von mehreren Servern bzw. das Wechseln des Servers während der Stream-Übertragung möglich.
- Das RSVP (Realtime Reservation Protocol) ist dafür zuständig, die Ressourcen (z.B. Bandbreite bei den Routern) auf dem Weg vom Sender zum Empfänger

derart zu reservieren, dass eine hinreichende Dienstgüte zugesagt werden kann. Das RSVP kann jedoch im bisherigen Internet nicht verwendet werden, da die üblichen Router eine Ressourcenreservierung nicht unterstützen.

Die vier beschriebenen Protokolle sind dabei nicht isoliert zu sehen, sondern arbeiten zur Auslieferung von Multimediastreams zusammen. RTSP baut dabei die Verbindung auf, während RTP und RTCP den Transport des Datenstroms übernehmen. Optional kann RSVP für den Stream eine „Ressourcenbuchung“ für den Transport organisieren.

Zur Zeit existieren im wesentlichen drei verbreitete Anbieter von Streaming-Technologien. Tabelle 1 illustriert die wichtigsten Anbieter mit ihrem Marktanteil:

Streaming-Technologie	Marktanteil
RealNetworks	ca. 50%
Windows MediaPlayer	30% - 40%
Apple Quicktime	ca. 9%

Tabelle 1: Verbreitete Anbieter von Streaming-Technologien [info02]

3 Forschungsstand

Peer-to-Peer Systeme zum asynchronen Ausliefern von Streaming Media wurden bisher vergleichsweise wenig diskutiert. Bezüglich dieses Themenkomplexes finden sich in der Literatur hauptsächlich technisch orientierte Beiträge [NgZa02a; NgZa02b]. So schlagen z.B. Xu, Hefeede, Hambrusch und Bhargava Algorithmen vor, die festlegen, welche Peers welche Abschnitte eines Streams ausliefern sollen [Xu+B02]. Auch die Firma CenterSpan (www.centerspan.com) stellt auf Ihren Webseiten ein System vor, welches verteiltes Streaming unterstützt [Cen02]. Mit PeerCast [Des+01] besteht ein System, welches verteiltes Streaming eines Livestreams ermöglicht. Hierbei handelt es sich also weniger um ein Streaming on-demand Modell, als vielmehr um die Verbreitung eines Streams, der synchron von vielen Konsumenten empfangen wird.

Der vorliegende Beitrag stellt ein mögliches Peer-to-Peer Streaming Geschäftsmodell vor, welches sich konzentriert auf die konzeptionell-ökonomische und weniger auf die technische Ebene. Eine solche wurde in der Literatur bisher kaum diskutiert.

4 Das Peer-to-Peer Geschäftsmodell

4.1 Begriffe

4.1.1 Peer-to-Peer

Ein Peer-to-Peer System besteht – im Gegensatz zu einem Client/Server System – aus vielen gleichartigen Knoten (Peers). Diese Peers interagieren, um eine bestimmte Aufgabe zu lösen. Dabei kann ein Peer temporär Client- bzw. Serverfunktionalitäten übernehmen. Die wohl bekanntesten Anwendungen, die gemäß dem Peer-to-Peer Paradigma aufgebaut sind, stellen Filesharing-Systeme [Leu02, S. 189ff] dar. Peer-to-Peer Systeme kommen in verschiedenen Architekturen vor, die sich im Grad der Dezentralisierung unterscheiden. Im Wesentlichen unterscheidet man Brokered Peer-to-Peer Systeme, Pure Peer-to-Peer Systeme und Hybride Peer-to-Peer Systeme [Min01;Min02]. Brokered Peer-to-Peer Systeme besitzen eine Zentraleinheit (Indexserver), die die Peers untereinander koordiniert. Der wesentliche Datenaustausch findet jedoch von Peer zu Peer statt. Für das vorliegende Geschäftsmodell kommt nur diese Architektur in Frage, da die Zentraleinheit Daten vorhalten muss, die nicht manipuliert werden dürfen (z.B. Kontostände oder kryptographische Schlüssel). Außerdem muss die Zentraleinheit Buch führen über die im Netz befindlichen Ressourcen, um Aufträgen entsprechende Ressourcen zuweisen zu können.

4.1.2 Geschäftsmodell

In der Literatur finden sich unterschiedliche Definitionen zum Geschäftsmodellbegriff. Für die folgenden Ausführungen wird der Geschäftsmodellbegriff von Wirtz verwendet: „Durch ein Geschäftsmodell wird in stark aggregierter Form abgebildet, welche Ressourcen in die Unternehmung fließen und wie diese durch den innerbetrieblichen Leistungsprozess in vermarktungsfähige Produkte und/oder Dienstleistungen transformiert werden.“ [Wirz01, S. 50]. Wirtz unterteilt ein Geschäftsmodell weiterhin in Partialmodelle wie z.B. das Leistungserstellungsmodell, das Erlösmodell, das Distributionsmodell oder das Finanzierungsmodell. Für die folgenden Ausführungen sind jedoch nur das Erlösmodell (wer bekommt welche Erlöse von wem?) und das Distributionsmodell (wie wird das Produkt – hier: der Stream – zum Konsumenten „ausgeliefert“) interessant, weil diese Partialmodelle sich von anderen traditionellen Geschäftsmodellen wesentlich unterscheiden.

4.2 Anforderungen an ein Streaming Media p2p Geschäftsmodell

Die Anforderungen an ein p2p basiertes Geschäftsmodell für die Auslieferung von Streams sind ähnlich wie die Anforderungen an kommerzielle p2p Filesharing-Systeme [GeAn02]. Das Ausliefern eines Streams stellt jedoch – im Vergleich zu einfachen p2p Down- bzw. Uploads - einen komplexeren Vorgang dar. Damit lassen sich folgende technische und ökonomische Anforderungen an ein p2p Streaming Media Geschäftsmodell formulieren:

- *Quality of Service.* Damit ein Stream unterbrechungsfrei ausgeliefert werden kann, muss die zur Verfügung stehende Bandbreite stets gewissen Mindestanforderungen genügen. Der Stream muss insofern fortlaufend mindestens „just in time“ bezogen werden können, damit keine unerwünschten Wartezeiten während des Streamkonsums auftreten. Dies ist ein entscheidender Unterschied zu herkömmlichen Downloads, bei denen eine solche Zeitrestriktion nicht beachtet werden muss. Diese Anforderung gewinnt an Wichtigkeit, als dass Konsumenten für den Stream zahlen und insofern auch eine einwandfreie Leistung erwarten.
- *Senkung von Infrastrukturkosten* durch Nutzung dezentraler Ressourcen. Streaming benötigt hohe und konstante Bandbreiten. Wird der Stream von vielen Konsumenten von einer Zentraleinheit bezogen, so erreichen die zentralen Streaming-Server schnell ihre Grenzen. Ein zentrales System skaliert insofern schlecht mit der Anzahl der Konsumenten. Zudem sind Investitionen in starke Server sehr kostenintensiv. Eine wichtige Anforderung ist, dass die Kosten eines p2p Streaming Systems geringer sind, als die Kosten zentraler Streaming-Server bzw. dass durch brachliegende dezentrale Ressourcen weitere Kapazitäten kostengünstig erschlossen werden können.
- *No free Riders / Entlohnung von Rechteinhabern.* Eine ähnliche Problematik wie auch bei konventionellen p2p Filesharing-Systemen ergibt sich auch in einem p2p Streaming System. Das Streaming in einem p2p System darf keine bloße Erweiterung herkömmlicher freier Tauschbörsen sein, sondern sollte als kommerzielles System die Entlohnung der Rechteinhaber berücksichtigen.
- *Content Control.* In einem p2p System “versorgen” sich die teilnehmenden Peers gegenseitig mit den nachgefragten Streams. Da es sich um ein kommerzielles System handelt, sollen nur diejenigen in den Genuss des Streams kommen, die diesen kostenpflichtig anfordern. Insofern dürfen Peers, die einen Stream ausliefern, nicht über diesen verfügen, wenn sie ihn nicht auch bezahlen. Ebenfalls sollte es nicht möglich sein, einen Stream während des Transports oder beim Zusammenfügen beim nachfragenden Peer abzugreifen. Hierfür sind Technologien des Digital Rights Management [Ros+02] und kryptographische Verfahren wie z.B. RSA [Riv+78] notwendig.

4.3 Konzeptioneller Aufbau des p2p Systems

Die Idee des hier vorgestellten p2p Geschäftsmodells beruht auf dem Gedanken, dass viel brachliegende Bandbreite „verteilt“ bei Tausenden von Endnutzern vorhanden ist. Es liegt insofern nahe, darüber nachzudenken, diese „dezentrale“ Bandbreite auszunutzen, um Streams verteilt auszuliefern (Bandwith-Sharing). Die Idee folgt dem Grundsatz „Teile und Herrsche“ und kann wie folgt zusammengefasst werden:

- Teile den Stream in viele kleine (vordefinierte) Fragmente auf. Die Fragmente sollten dabei am besten gleich groß sein und eine gleiche Abspieldauer haben. Zwar kann es möglich sein, dass ein Fragment aufgrund von Bandbreitenengpässen von mehreren Peers gleichzeitig bezogen werden muss (Multipler Download). Die Ausprägung dieser „Subfragmente“ ist jedoch nicht vordefiniert und hängt von der temporären Auslastung der bereitstellenden Peers ab.
- Bringe die unterschiedlichen Fragmente redundant auf sehr vielen Peers unter.
- Möchte jemand einen Stream konsumieren, so koordiniere einen verteilten „Just-in-Time“ Download der Fragmente von verschiedenen Peers, so dass der Stream letztendlich wieder zusammengesetzt werden kann.

Eindeutiges Ziel einer solchen Vorgehensweise ist das Einsparen kostenintensiver zentraler Infrastrukturen zugunsten einer koordinierten Nutzung verteilter brachliegender Ressourcen („Koordination statt Investition“).

Zur Koordination der Fragmentdownloads ist eine Zentraleinheit notwendig, die Download-Pläne erstellt, welche regeln, welches Fragment zu welchem Zeitpunkt von welchem Peer von welchem Konsumenten bezogen wird. Diese Zentraleinheit hat eine vermittelnde Funktion und ähnliche Aufgabe wie der Indexserver im einstigen p2p-System Napster [Shir01]. Abbildung 2 illustriert den Vorgang eines verteilten Streamdownloads.

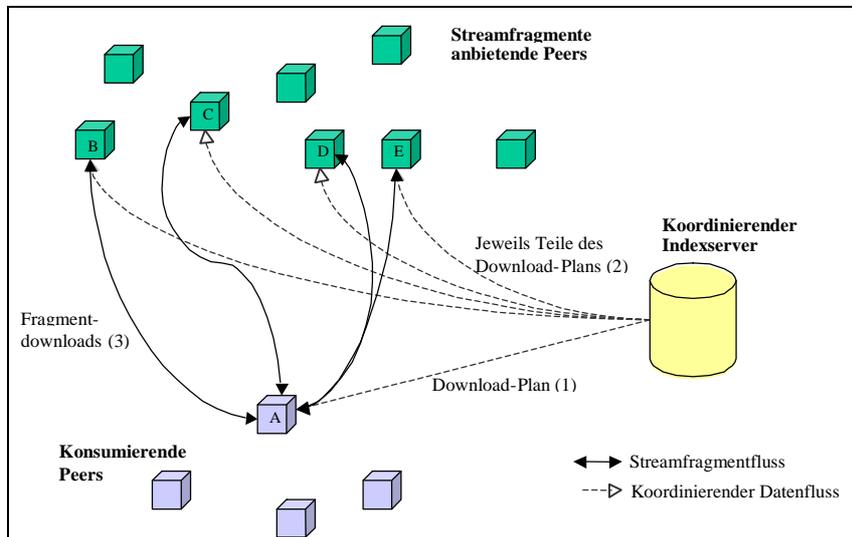


Abbildung 2: Koordination des verteilten Downloads eines Streams

Die Koordination eines verteilten Streamdownloads kann in drei Schritten durchgeführt werden:

1. Zunächst meldet ein Peer (hier A) dem Indexserver, dass er einen bestimmten Stream empfangen möchte. Der Indexserver ermittelt in seiner Datenbank, welche Peers verfügbar sind, die entsprechende Fragmente des Peers bereithalten. Aus diesen Informationen kann der Indexserver – unter Berücksichtigung bereits ausgegebener Downloadpläne - einen vorläufigen Downloadplan an Peer A senden. Dieser Downloadplan enthält einfach eine Liste, aus der hervorgeht, welches Fragment von welchem Peer in welcher Reihenfolge herunterzuladen ist. Der Downloadplan muss derart gestaltet werden, so dass Peers bzw. Fragmente möglichst keinen Engpass darstellen.
2. In einem zweiten Schritt werden die betreffenden Peers darüber informiert, dass bei ihnen entsprechende Uploads anstehen. Dies soll vor allem dem Umstand vorbeugen, dass die Besitzer der Peers nicht einfach abschalten, obwohl noch Uploads geplant sind. Damit unvorhergesehenes Abschalten eines Peers ohne Abmeldung aus dem p2p-System möglichst nicht vorkommt, müssen Sanktionsmechanismen implementiert werden (z.B. schlechte Bewertung oder monetäre Strafen).
3. Im dritten Schritt lädt das Peer A die Fragmente sukzessive von den anbietenden Peers B,C,D und E herunter. Die Reihenfolge wurde durch den Download-Plan festgelegt. In der Abbildung ist nicht dargestellt, dass ein Fragment aus Bandbreitenengpassgründen auch von mehreren Peers gleichzeitig heruntergeladen werden kann (Download-Splitting eines Fragments).

Der beschriebene Ablauf des verteilten Ausliefern eines Streams stellt das System während des laufenden Betriebs dar. Dabei sind zwei Aspekte zu beachten:

Der erste Aspekt zielt darauf ab, wie die initiale Verteilung der Fragmente funktioniert. Aufgrund der Verteiltheit des p2p Systems, ist es nicht notwendig, dass alle auszustattenden Peers die Fragmente von einer Zentralinstanz – wie bspw. dem Indexserver – herunterladen. Die Ausstattung eines Peers mit einem Fragment kann wiederum durch einen Download an einem anderen Peer geschehen. Ein Fragment diffundiert insofern von einem Peer zum anderen, was wiederum dezentral vorhandene Netzwerkbandbreite ausnutzt. Alle Fragmente eines Streams müssen nur einmal durch ein „Seed-Peer“ in das p2p System eingebracht werden. Die anschließende notwendige redundante Verteilung der Fragmente geschieht daraufhin von Peer zu Peer. Um sicherzustellen, dass die Fragmente für fragment anbietende Peers unbrauchbar sind, bietet es sich an, die Fragmente verschlüsselt auf den Peers unterzubringen. Immerhin sollen die fragment anbietenden Peers die Fragmente lediglich ausliefern, nicht aber selbst nutzen können. Um Manipulationen von Fragmenten auszuschließen, können Mechanismen wie digitale Fingeraabdrücke zum Einsatz kommen.

Der zweite Aspekt adressiert den Problemkomplex, dass Fragmentengpässe auftreten können, nachdem schon Download-Pläne an Konsumenten ausgeliefert wurden. Beispielsweise ist es möglich, dass eingeplante Peers während der Auslieferung eines Streams an einen Konsumenten sich abmelden und diese insofern nicht mehr für die Auslieferung von Fragmenten zur Verfügung stehen. Das Bemerkens eines solchen Engpasses ist aufgrund der Download-Pläne möglich, bevor der Engpass wirksam wird und Störungen verursacht. Eine Engpasssituation kann zwei Folgen haben:

Sind genug andere Peers mit den betreffenden Fragmenten im p2p System, so ist es lediglich notwendig, die Download-Pläne der betreffenden Konsumenten neu zu strukturieren. Die adaptierten Download-Pläne müssen dann an die betreffenden Peers verschickt werden.

Liegt jedoch der Fall vor, dass ein tatsächlicher Engpass vorliegt, weil zu wenige Fragmente verfügbar sind, um die Nachfrage zu befriedigen, so muss eine spontane Replikation betreffender Fragmente geschehen, bevor das Fragment tatsächlich benötigt wird. Wie auch bei der initialen Verteilung von Fragmenten geschieht auch dies wieder von Peer zu Peer und wird durch den Indexserver koordiniert.

4.4 Erlösmodell

Das vorgestellte p2p System wäre kein Geschäftsmodell, wenn nicht auch Erträge mit einem solchen System erwirtschaftet werden könnten. Insofern stellt sich die Frage, wie der Infrastrukturbetreiber (also der Betreiber des Indexservers bzw. die Entwickler der Peer-Software) entlohnt werden können. Neben dem Infrastruktur-

betreiber müssen auch den Peers Anreize geboten werden, Fragmente zu speichern und auszuliefern. Da das zur Verfügungstellen von Speicherplatz und Netzwerkbandbreite durch die Peers aus betriebswirtschaftlicher Sicht eine Bereitstellung von Produktionsfaktoren darstellt, liegt es nahe, die Peers für das Ausliefern von Fragmenten zu entlohnen.

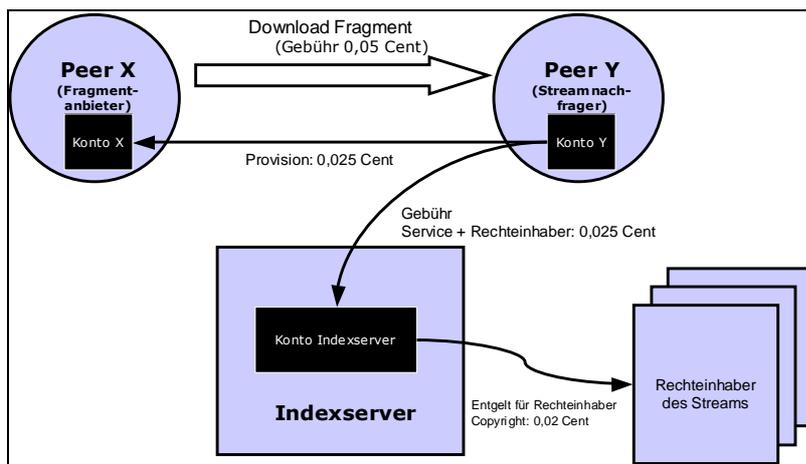


Abbildung 3: Entlohnung der Beteiligten beim Ausliefern eines Streams

Abbildung 3 zeigt, welche Beteiligten wofür eine Entlohnung erhalten. Peer Y lädt gemäß Download-Plan ein bestimmtes Fragment von Peer X herunter. Aufgrund des Downloads muss Peer Y insgesamt z.B. 0,05 Cent bezahlen. Für diesen Bezahlvorgang existieren auf dem Indexserver Konten der jeweiligen Peers, auf denen Beträge verbucht werden können. Die 0,05 Cent für den Fragmentdownload werden anschließend aufgeteilt: ein Betrag von 0,025 (also 50%) erhält Peer X, weil er das Fragment ausgeliefert hat. Die restlichen 0,025 Cent werden zwischen dem Rechteinhaber des Streams (0,02 Cent) und dem Infrastrukturbetreiber (0,005 Cent) aufgeteilt. Als Alternative zur Bepreisung von einzelnen Fragmenten kann auch eine prozentuale Bepreisung gemessen am Preis des Gesamtstreams dienen. Die Entlohnung von fragment anbietenden Peers stellt einen Anreiz dar, unausgenutzte Ressourcen zur Verfügung zu stellen. Die Abrechnung wickelt stets der Indexserver ab. Die Konten der Peers müssen zuvor durch Einzahlungen aufgefüllt werden, wenn ein kostenpflichtiger Stream konsumiert werden soll. Die in Abbildung 3 dargestellten Konten sind dabei aus logischer Sicht abgebildet. Die tatsächliche Verwaltung findet natürlich auf dem Indexserver statt. Fragment anbietende Peers sollten dabei selbst entscheiden können, welche Streams sie ausliefern wollen. Hierfür sollte die Peer-Software Funktionalitäten bieten, um z.B. anzugeben, ab welchem Preis pro Kbyte man bereit ist, ein Fragmentstream anzubieten. Mit einem solchen Marktmechanismus wäre eine dynamisch Allokation von dezentralen Ressourcen möglich.

5 Ressourcenanalyse des Geschäftsmodells

5.1 Modell zur Ressourcenanalyse

Im vorangegangenen Abschnitt wurde die konzeptionelle Architektur bzw. das Erlösmodell des Peer-to-Peer basierten Geschäftsmodell dargestellt. An dieser Stelle sollen Überlegungen angestellt werden, wie viele Peers benötigt werden, um den Empfängern der Streams einen störungsfreien Konsum zu ermöglichen. Auf wie vielen Peers die Fragmente untergebracht werden müssen, hängt dabei von zwei Engpassfaktoren ab:

- *Die verfügbare Bandbreite als Engpass.* Jedes Peer stellt eine gewisse Bandbreite zur Verfügung, um die Konsumenten mit Fragmenten zu versorgen. Da in der Regel mehrere Konsumenten einen Stream gleichzeitig konsumieren¹, ist es notwendig, Fragmente redundant zu halten, damit keine nicht zu befriedigende „Übernachfrage“ nach Bandbreite entsteht und somit Störungen während des Streams auftreten.
- *Die Verfügbarkeit von Fragmenten als Engpass.* Neben der Bandbreite ist die Verfügbarkeit jedes Peers, auf dem die Fragmente liegen, grundsätzlich wahrrscheinlichkeitsbehaftet. Es ist deshalb auch aus diesem Grund notwendig, Fragmente redundant zu halten, da eine gewisse Mindestverfügbarkeit jedes Fragments gewährleistet werden muss.

Damit der Stream möglichst ohne Störung übertragen werden kann, müssen beide Engpassarten berücksichtigt werden.

Die Bandbreite als Engpass

Für die Analyse der Engpässe ist es zunächst notwendig, einige Annahmen zu treffen. Obwohl diese Annahmen stark von der Realität abstrahieren, können dennoch einige grundsätzliche Erkenntnisse gewonnen werden. Die Annahmen stellen sich wie folgt dar:

- Jedes Peer stellt eine konstante Bandbreite zur Verfügung, die sich die Konsumenten „teilen“ müssen.

¹ Mit dem gleichzeitigem Konsum ist nicht gemeint, dass alle Konsumenten gemeinsam den Stream gleichzeitig an der selbe Stelle benötigen (Live-Stream). Vielmehr ist dabei gemeint, dass die Konsumenten den Stream an jeweils verschiedenen Stellen benötigen (Video-on-Demand / Music-On-Demand), je nachdem, zu welcher Zeit sie den Stream angefordert haben.

- Jedes Peer ist mit einer konstanten Wahrscheinlichkeit verfügbar.

Die zu beantwortende Frage ist: „Wie viele Peers werden mindestens benötigt, damit die Bandbreite für alle Konsumenten ausreicht, sodass keine Störungen während des Konsums des Streams auftreten?“.

Für den Modellrahmen werden folgende Parameter benötigt:

b = zur Verfügung stehende Bandbreite jedes Peers in [Byte/Sekunde]

p = Wahrscheinlichkeit, dass das Peer verfügbar ist

M = Anzahl der unterschiedlichen Fragmente, in die der Stream aufgeteilt wurde

v = Anzahl der Konsumenten, die den Stream gleichzeitig empfangen

s = Größe eines Fragmentes in Byte

N = Anzahl benötigten Peers insgesamt

R* = Gesamtredundanz pro Fragment (jedes Fragment wird insgesamt R* mal auf verschiedenen Peers untergebracht)

T = Gesamtabspieldauer des Streams

Merkmal eines Streams ist, dass das Abspielen des Streams mindestens zeitgleich mit dem fortlaufenden Download einhergeht. Hierdurch kann ein Stream „Just in Time“ abgespielt werden. Voraussetzung ist allerdings, dass die Bandbreite ausreicht, um den Stream unterbrechungsfrei empfangen und abspielen zu können. Im Folgenden wird davon ausgegangen, dass der Streamempfänger ausreichend Bandbreite besitzt, um den Stream unterbrechungsfrei zu empfangen. Wird die abzuspielende Datei in einem Peer-to-Peer Netzwerk in Fragmente geteilt und dezentral untergebracht, so muss der Stream von verschiedenen Peers heruntergeladen und zusammengesetzt werden. Dabei ist damit zu rechnen, dass ein Peer nicht in der Lage ist, ein Streamfragment ausreichend schnell zu liefern, da möglicherweise nur eine schwache Netzwerkanbindung vorhanden ist. Um eine hinreichende Bandbreite zu erhalten, muss der Download eines Fragmentes somit parallel von mehreren Peers erfolgen (Download-Splitting). Zunächst ist also wichtig zu ermitteln, wie viele Bandbreiten eines Peers ein Konsument parallel in Anspruch nehmen muss, um ein Fragment herunterzuladen. Die Downloadzeit muss, um einen „Just in Time“ Stream zu erhalten, somit der Abspieldauer des Fragmentes entsprechen, so dass folgende Bedingung erfüllt werden muss:

$$k \cdot \frac{s}{b} = \frac{T}{M} \Leftrightarrow k = \frac{b \cdot T}{s \cdot M} \quad (1)$$

k entspricht dabei der Anzahl der gleichzeitig für den Download in Anspruch genommenen Peers aus Sicht eines einzelnen Konsumenten, um eine hinreichende Bandbreite zu erreichen (ist k kleiner 1, so bedeutet dies, dass ein Peer mehr als einen Konsumenten gleichzeitig versorgen kann).

Die benötigte Gesamtanzahl an Peers ist das Produkt aus benötigten Peers pro Fragmentdownload und der Anzahl der Konsumenten:

$$K = k \cdot v \quad (2)$$

K ist dabei die Gesamtanzahl an verfügbaren Peers, sodass die Bandbreite für alle v Konsumenten ausreicht. K stellt ein Mindestmaß an Bandbreite dar. Die Peers sind jedoch nur mit der Wahrscheinlichkeit p online und verfügbar. Es stellt sich somit die Frage, wie viele Peers N insgesamt vorhanden sein müssen, damit mit einer hohen Wahrscheinlichkeit mindestens K Peers verfügbar sind. Die Verfügbarkeit von Peers kann als ein „Ziehen mit Zurücklegen“ interpretiert werden und gehorcht insofern einer Binomialverteilung [BaBa96, S.100]. Damit mindestens K Peers mit einer hohen Wahrscheinlichkeit α (z.B. 0,99) verfügbar sind, muss also N so bestimmt werden, dass gilt:

$$\sum_{x=K}^N \binom{N}{x} p^x (1-p)^{N-x} \geq \alpha \quad (3)$$

Gleichung (3) ist die kumulierte Dichtefunktion der Binomialverteilung und entspricht somit ihrer Verteilungsfunktion.

Geht man von einer hohen Anzahl Konsumenten v aus, so nehmen damit auch die Werte von K (Mindestanzahl an Peers) und N (Potenziell verfügbare Peers) zu. Bei hinreichend großem N lässt sich die Binomialverteilung aufgrund des Grenzwertsatzes von DeMoivre-Laplace [Schl96, S. 94] durch eine Normalverteilung wie folgt approximieren:

$$\binom{N}{x} p^x (1-p)^{N-x} = \frac{1}{\sqrt{2\pi N p(1-p)}} \cdot e^{-\frac{(x-Np)^2}{2Np(1-p)}} \quad (4)$$

Eine genügende Annäherung kann angenommen werden, wenn gilt $pN > 10$ und $N(1-p) > 10$ [Vo00, S.359]. Da stets $K \leq N$ und K proportional zu der Anzahl der Konsumenten v ist, erreicht allein K bereits bei $v=100$ einen ausreichend hohen Wert, sodass die Näherung akzeptiert werden kann.

Die Bedingung für die Anzahl der potenziell verfügbaren Peers N aus (3) ist somit durch Gleichung (4) approximierbar. Somit ist N so zu bestimmen, so dass gilt:

$$\int_K^N \frac{1}{\sqrt{2\pi N p(1-p)}} \cdot e^{-\frac{(x-Np)^2}{2Np(1-p)}} dx = \alpha \quad (5)$$

Gleichung (5) lässt sich leichter handhaben, da die Berechnung großer Binomialkoeffizienten entfällt und für die Normalverteilung auf die üblichen Wahrscheinlichkeitstabellen zurückgegriffen werden kann.

Fragment als Engpass

Ein bestimmtes Fragment ist nicht zwangsweise verfügbar, da die Peers nur mit einer bestimmten Wahrscheinlichkeit p online sind. Es muss jedoch sichergestellt werden, dass jedes Fragment mit einer gewissen hohen Wahrscheinlichkeit verfügbar ist. Hierfür ist es notwendig, dass Fragmente redundant auf verschiedenen Peers gespeichert werden. Für die Untersuchung des Engpasses der Verfügbarkeit eines Fragmentes wird analog zum Engpass Netzbandbreite die obige Analyse fortgesetzt.

Der Bezug eines einzelnen Fragmentes unterliegt zwei Engpässen. Einerseits ist es möglich, dass ein Fragment zur gleichen Zeit von mehreren Konsumenten benötigt wird. Damit die Bandbreite hierfür ausreicht, muss das Fragment somit redundant auf mehreren Peers vorgehalten werden. Unterstellt man, dass der Zeitpunkt der Anforderung des Streams durch die Konsumenten gleichverteilt ist, so ist die Wahrscheinlichkeit, dass ein Konsument ein bestimmtes Fragment zu einem gegebenen Zeitpunkt benötigt $\frac{1}{M}$ (M ist die Anzahl der verschiedenen Fragmente, in

die der Stream aufgeteilt wird). Damit keine Bandbreitenengpässe bezogen auf ein einzelnes Fragment auftreten, sollte ein Fragment in der Form redundant gehalten werden, so dass diese Redundanz mit einer hohen Wahrscheinlichkeit ausreicht, um die Nachfrage der Konsumenten zu befriedigen. Mit Hilfe der Binomialverteilung lässt sich somit die Frage beantworten: Wie viele der v Konsumenten benötigen zur gleichen Zeit ein bestimmtes Fragment mit einer hohen Wahrscheinlichkeit β ? Somit sollte die Redundanz pro Fragment R durch folgende Bedingung ermittelt werden:

$$\sum_{x=0}^R \binom{v}{x} \left(\frac{1}{M}\right)^x \left(1 - \frac{1}{M}\right)^{v-x} \geq \beta \quad (6)$$

Mit Hilfe von Gleichung (6) kann man die Redundanz pro Fragment berechnen. Da ein Fragment aufgrund zu geringer Bandbreite eines Peers von mehreren Peers gleichzeitig bezogen werden muss, müssen mindestens $R \cdot k$ verfügbare Peers das Fragment bereitstellen.

Bisher wurde lediglich diskutiert, wie viele verfügbare Peers das Fragment bereitstellen müssen. Jedoch ist die Verfügbarkeit der Peers wahrscheinlichsbehaftet. Insofern müssen grundsätzlich weitere Peers das Fragment besitzen, damit mit einer hohen Wahrscheinlichkeit ausreichend Peers mindestens verfügbar sind. Die Berechnung der Gesamtredundanz R^* muss also unter Einbezug der Online-Wahrscheinlichkeit p geschehen. Die Redundanz R^* sollte so hoch sein, dass un-

ter Einbezug der stochastischen Verfügbarkeit der Peers *und* etwaiger paralleler Zugriffe auf ein Fragment eine möglichst kleine Wahrscheinlichkeit besteht, dass es einen Fragmentengpass gibt. Diese Fragestellung lässt sich beantworten, indem die Wahrscheinlichkeiten von Situationen, die zu einem Fragmentengpass führen, aufaddiert werden und die Summe dieser Wahrscheinlichkeiten hinreichend klein bleibt. Dabei wird davon ausgegangen, dass die Wahrscheinlichkeit der Verfügbarkeit eines Peers p und die Wahrscheinlichkeit, dass ein Fragment gleichzeitig von i Konsumenten benötigt wird, unabhängig voneinander sind. Folgendes Zahlenbeispiel erläutert die Vorgehensweise:

Es sei $p=0,4$ (Onlinewahrscheinlichkeit) ; $v=2500$ (Anzahl Konsumenten); $M=10000$ (Anzahl verschiedener Fragmente); $k=3$ (parallel benötigte Peers für Download eines Fragments).

Es liegt nun eine Engpasssituation, wenn z.B. folgendes Eintritt:

Ein Fragment wird gleichzeitig von 2 Konsumenten gebraucht. Da $k=3$ ist, werden hierfür 6 Peers gleichzeitig benötigt. Ein Engpass entsteht, wenn nur bis zu 5 Peers das gewünschte Fragment bieten können. Aufgrund der Unabhängigkeit ergibt sich die Wahrscheinlichkeit für diesen Engpass aus dem Produkt der Wahrscheinlichkeiten, dass genau zwei Konsumenten das Fragment zur gleichen Zeit brauchen und dass maximal 5 verfügbare Peers das Fragment bieten können. In folgender Tabelle sind alle für die Beispielzahlen möglichen Engpässe aufgeführt:

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Anzahl Fragm., die gleichz. benötigt werden	Hierfür benötigte Peers (k-fache)	Wkt., dass genau diese Anzahl Fragmente benötigt wird	höchstens verfügbare Peers: (2)-1	Wahrscheinlichkeit, dass höchstens (4) Anzahl Peers verfügbar sind (hängt von R* ab)	Engpasswahrscheinlichkeit: (3)*(5)	Kumulierte Wahrscheinlichkeiten
1	3	19,4717%	2	0,0429%	0,0084%	0,0084%
2	6	2,4332%	5	2,9362%	0,0714%	0,0798%
3	9	0,2026%	8	27,3531%	0,0554%	0,1352%
4	12	0,0127%	11	73,2282%	0,0093%	0,1445%
5	15	0,0006%	14	96,5608%	0,0006%	0,1451%
6	18	0,0000%	17	99,8795%	0,0000%	0,1451%
7	21	0,0000%	20	99,9992%	0,0000%	0,1451%
8	24	0,0000%	23	100,0000%	0,0000%	0,1451%

Tabelle 2: Fragmentengpassbetrachtung. Es wurde eine Gesamtredundanz R*=25 zugrundegelegt. Mit einer Wahrscheinlichkeit von 0,145% tritt ein Engpass auf.

Als allgemeine Bedingung für die Ermittlung von R* kann folgende Gleichung gelten:²

$$\sum_{i=1}^{\infty} \left(\underbrace{BV\left(i, v, \frac{1}{M}\right)}_{\text{Wahrscheinlichkeit, dass } i \text{ Konsumenten das Fragment benötigen}} \cdot \underbrace{\sum_{j=0}^{\text{aufrunden}(ki)-1} BV(j, R^*, p)}_{\text{Wahrscheinlichkeit, dass ein Engpass entsteht}} \right) \leq \varepsilon \quad (7)$$

Dabei entspricht die Funktion BV(x,y,z) der Dichtefunktion der Binomialverteilung mit der Wahrscheinlichkeit z, der Anzahl Ziehungen y und der positiven Ziehungen x. R* muss nun so gewählt werden, dass die Engpasswahrscheinlichkeit einem möglichst kleinem Niveau ε entspricht.

² Die Summe muss theoretisch bis unendlich berechnet werden. Dabei muss beachtet werden, dass der rechte Ausdruck (Wahrscheinlichkeit, dass ein Engpass eintritt) stets als 1 interpretiert werden muss, sobald j>R*.

5.2 Ein Zahlenbeispiel

Um die theoretisch hergeleiteten Ressourcenumfänge plastischer darzustellen, sollen an dieser Stelle zwei Beispiele erfolgen. Dabei werden zwei Szenarien betrachtet. Im ersten Szenario besitzen die Peers nur wenig Bandbreite, im zweiten dagegen verhältnismäßig viel. Folgende Tabelle zeigt die Konfigurationen der beiden Szenarien:

Parameter	Ausprägung
Anzahl Konsumenten v	2500
Anzahl unterschiedlicher Fragmente M	10000
Größe eines Fragments s [byte]	52429
Größe des Gesamtstreams [Mbyte]	500
Spieldauer des Streams T [sec]	3600 [=60 min]
Peer Online Wahrscheinlichkeit p	40%

Tabelle 3: Parameterkonstellation der beiden Szenarien

Im Szenario 1 (wenig Bandbreite) soll eine Übertragungskapazität von 37500 [Byte/Sec] angenommen werden. Das entspricht ca. einer 5-fachen ISDN Leitung. Im Szenario 2 wird eine Kapazität von 1 Mio. [Byte/Sec] angenommen. Bei dieser Parameterkonstellation sind gemäß Gleichung (5) folgende Anzahlen an Peers insgesamt notwendig, um die erforderliche Bandbreite bei einem Verfügbarkeitsniveau α von z.B. 99% zu gewährleisten:

Parameter	Szenario 1 (wenig Bandbreite)	Szenario 2 (viel Bandbreite)
Benötigte Peers insgesamt	24721 Peers	1000 Peers
Benötigte parallele Peers k	3,883	0,146

Tabelle 4: Notwendige Anzahl Peers in den beiden Szenarien

Tabelle 4 stellt dar, wie viele Peers benötigt werden, um in den Szenarien eine hinreichende Bandbreite zu erreichen. Folgende Tabellen beschäftigen sich mit der Redundanz R^* der Fragmente in den Szenarien. Die folgende Tabelle 5 zeigt analog zu Tabelle 2, wie R^* im Szenario 2 (viel Bandbreite) ermittelt werden kann:

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Anzahl gleichzeitig angeforderter Fragmente	Hierfür benötigte Peers (k-fache aufgerundet)	Wahrscheinlichkeit, dass genau diese Anzahl Fragmente benötigt wird	Höchstens verfügbare Peers	Wahrscheinlichkeit, dass höchstens diese Anzahl Peers verfügbar sind	Engpasswahrscheinlichkeit	Kummulierte Wahrscheinlichkeiten
1	1	19,4717%	0	1,6796%	0,3271%	0,3271%
2	1	2,4332%	0	1,6796%	0,0409%	0,3679%
3	1	0,2026%	0	1,6796%	0,0034%	0,3713%
4	1	0,0127%	0	1,6796%	0,0002%	0,3715%
5	1	0,0006%	0	1,6796%	0,0000%	0,3715%
6	1	0,0000%	0	1,6796%	0,0000%	0,3715%
7	2	0,0000%	1	10,6376%	0,0000%	0,3715%
8	2	0,0000%	1	10,6376%	0,0000%	0,3715%
9	2	0,0000%	1	10,6376%	0,0000%	0,3715%
10	2	0,0000%	1	10,6376%	0,0000%	0,3715%
11	2	0,0000%	1	10,6376%	0,0000%	0,3715%
12	2	0,0000%	1	10,6376%	0,0000%	0,3715%
13	2	0,0000%	1	10,6376%	0,0000%	0,3715%
14	3	0,0000%	2	31,5395%	0,0000%	0,3715%
15	3	0,0000%	2	31,5395%	0,0000%	0,3715%
...

Tabelle 5: Fragmentengpasskalkulation im Szenario (2) viel Bandbreite ($R^*=8$)

In Tabelle 5 wird jedes Fragment $R^*=8$ mal redundant gehalten. Es gibt insofern grundsätzlich 8 verschiedene Peers, die dasselbe Fragment bereithalten. Da ein Peer soviel Bandbreite hat, dass zur selben Zeit mehrere Fragmente gleichzeitig abgerufen werden können, werden erst in der Situation, in der ein Fragment 7 mal gleichzeitig benötigt wird, zwei Peers zum Ausliefern notwendig (Spalte 2). In Tabelle 5 tritt bei $R^*=8$ ein Engpass mit einer Wahrscheinlichkeit von 0,37% auf.

Eine Vergleichbare Rechnung ist auch für Szenario 2 möglich. Ohne auch diese Rechnung detailliert dazulegen kann für die beiden Szenarien folgendes Ergebnis berechnet werden:

Parameter	Szenario 1 (wenig Bandbreite)	Szenario 2 (viel Bandbreite)
Redundanz pro Fragment R*	27	8
Engpasswahrscheinlichkeit eines Fragmentes	0,389%	0,371%
Verschiedene Fragmente pro Peers im Mittel	$27 \cdot 10000 / 24721 = 1,092$	$8 \cdot 10000 / 1000 = 80$

Tabelle 6: Verschiedene Gesamtredundanzen R* in den beiden Szenarien

An Tabelle 6 kann man erkennen, dass eine größere Bandbreite der Peers mit einer geringeren Redundanz einhergeht. Dies lässt sich vor allem dadurch begründen, dass ein Peer stets mehrere Fragmente gleichzeitig aufgrund der größeren Bandbreite ausliefern kann. Es sind somit weniger Peers notwendig, um die gleichzeitige Nachfrage nach einem Fragment zu befriedigen. In der letzten Zeile ist dargestellt, wie viele verschiedene Fragmente ein Peer im Mittel bereitstellen muss. Je höher die Bandbreite der Peers, desto mehr verschiedene Fragmente muss es beithalten.

Die verschiedenen (redundanten) Fragmente müssen auf die gesamten Peers aufgeteilt werden. Bei der Aufteilung sollte möglichst vermieden werden, dass der Ausfall eines bestimmten Fragmentes mit dem Ausfall eines bestimmten anderen Fragmentes korreliert. Dies gilt verschärft für Fragmente, die in der Abspielfolge zeitlich nah aneinander liegen.

6 Ökonomische Analyse

Im letzten Abschnitt wurden einige Überlegungen dazu angestellt, inwiefern der Stream auf die verschiedenen Peers aufgeteilt werden muss und welche Einflussfaktoren eine Rolle dabei spielen. Nach dieser eher technisch geprägten Analyse sollen nun einige Argumente dargestellt werden, die die ökonomische Vorteilhaftigkeit einer derart verteilten Peer-to-Peer Anwendung herausstellen. Im wesentlichen können hier drei verschiedene ökonomische Wirkungen identifiziert werden: Kosten und Investitionen, Verwendung marktlicher Koordinationsmechanismen und Wirkungen auf Eintrittsbarrieren.

Kosten und Investitionen

Der Grundgedanke der vorgestellten p2p-Applikation bzw. des Geschäftsmodells ist grundsätzlich nach folgendem Prinzip aufgebaut: Nutze dezentrale Ressourcen, die ohnehin vorhanden sind und verknüpfe sie zu einer für den Zweck der Anwendung brauchbaren Infrastruktur. Hiermit wird deutlich, dass Investitionen in eine eigene aufzubauende Infrastruktur nur in geringem Maße notwendig sind. Wie anfangs bereits beschrieben, setzen Streaming-Anwendungen enorme Bandbreiten und Speicherplatzmöglichkeiten voraus. Auch der Aufbau einer lastverteilenden und somit redundanten Infrastruktur ist notwendig, wenn die Bereitstellung der Inhalte durch einen eigenen Maschinenpark bewältigt werden soll. Diese Infrastrukturkosten können bei dem vorgestellten p2p Geschäftsmodell größtenteils entfallen. Es ist lediglich notwendig, die Kosten für die Entwicklung der Peer-Software und der koordinierenden Zentraleinheit zu bewältigen. Infrastrukturkosten in Form von vielen starken Servern sind kaum mehr vorhanden.

Bandbreite und Speicherplatz werden dezentral bei vielen tausenden Usern „angemietet“. Ökonomisch gesehen bedeuten diese „Mieteinnahmen“ für den einzelnen User die Vermeidung von Leerkosten, da jeder User seine Hardware in der Vergangenheit anschaffen musste. Insofern besteht zwischen jedem User und dem Betreiber der dezentralen p2p-Infrastruktur ein Win-Win-Verhältnis. User, die intensiv Fragmente speichern und bereitstellen könnten sogar umsonst in den Genuss der Streams kommen, wenn sich ihre „Mieteinnahmen“ und die zu zahlenden Preise für die konsumierten Streams kompensieren.

Neben der nicht mehr notwendigen Investition in eine zentrale Infrastruktur weist das dezentrale p2p-Modell auch Vorteile im „laufenden Betrieb“ auf. Die User, die auf ihren Peers Fragmente zur Verfügung stellen, werden nur entlohnt, wenn tatsächlich auch Uploads durchgeführt werden. Insofern entstehen auch nur Kosten bei Nutzung des Systems. Dies bedeutet eine sehr variable Kostenskalierung. Der Break-Even eines solchen Systems würde aufgrund der Variabilisierung der Kosten eher erreicht als beim Vorhalten zahlreicher Streaming-Server. Mit dieser Vermeidung von Leerkosten auf Seiten des Infrastrukturbetreibers geht eine Risikoverminderung einher.

Verwendung marktlicher Koordinationsmechanismen

Ist ein bestimmter Stream sehr beliebt (z.B. ein neu herausgekommener Film), so sind aufgrund der gestiegenen Nachfrage mehr Peers notwendig, die Fragmente dieses Films bereitzuhalten. Aufgrund der Beliebtheit des Streams wird auch die Zahlungsbereitschaft der Konsumenten entsprechend höher sein. Wird die erhöhte Zahlungsbereitschaft in Teilen an die User weitergegeben, die auf ihrem Peer Fragmente hosten und dafür entlohnt werden, so entstehen erhöhte Anreize, gerade Fragmente dieses Streams anzubieten. Insofern führt eine erhöhte Nachfrage über die Entlohnung der bereitstellenden Peers zu einer Erweiterung der verfügbaren Ressourcen, die wiederum notwendig sind, die erhöhte Nachfrage mit ausreichend Bandbreite zu versorgen. Diese marktliche Koordination führt also tenden-

ziell zu einer geeigneten und situationsgerechten Skalierung der Ressourcen, mit hin also zur jeweils ausreichenden Allokation der notwendigen Produktionsfaktoren, um den Stream insgesamt mehrfach auszuliefern. Insofern kann hierbei von einer Selbstallokation der Ressourcen durch den Preismechanismus gesprochen werden. Dennoch muss die Kritik angeführt werden, dass das p2p-System ein gewisses Mindestmaß an Teilnehmern (kritische Masse) aufweisen muss, damit es funktionstüchtig ist. Für diesem Zweck ist es von Nutzen, die Streaming-Funktionalitäten an bereits weitverbreitete p2p Filesharing-Systeme zu koppeln.

Eintrittsbarrieren

Die Speicherung der Streams geschieht im vorgestellten p2p-System dezentral und redundant. Da keine Zentralinstanz die Streams ausschließlich vorhält und die Kosten hochgradig mit der Nutzung eines Streams skalieren, könnte das p2p-System Funktionalitäten bereithalten, die es praktisch jedem User ermöglichen, eigene Streams kostenpflichtig über das p2p-System zu publizieren. Auch unbekannte „Produzenten“ könnten somit kostengünstig in direkten Kontakt mit Konsumenten treten und Eintrittsbarrieren würden insofern stark gesenkt. Es ist somit denkbar, einen Markt für Hobby- und Privatinhalte zu schaffen, der ansonsten aus Kostengründen nicht möglich wäre. Hierbei muss allerdings kritisiert werden, dass die inhaltliche Kontrolle der Streams ein ernstzunehmendes Problem darstellt. So darf es bspw. nicht dazu kommen, dass Streams, die normalerweise unter eine Zensur fallen würden, öffentlich zugänglich werden. Um solche Situationen zu vermeiden, müssen die Urheber der Streams eindeutig identifizierbar sein, damit diese im Zweifelsfall zur Verantwortung gezogen werden können.

7 Zusammenfassung

Der vorliegende Beitrag stellte ein p2p basiertes Geschäftsmodell vor, welches die dezentrale Auslieferung von Multimediasstreams fokussiert. Dieses Geschäftsmodell weist dabei eine andere Kostenstruktur auf als ein Ausliefern von Streams mithilfe zentraler Streaming-Server. Die Ressourcenanalyse hat gezeigt, wie viele Peers notwendig sind, um ausreichend Bandbreite vorzuhalten. Diese Analyse basierte jedoch auf einfachen Annahmen. Insbesondere die Annahme, dass alle Peers über eine gleiche Bandbreite verfügen, kann als sehr simplifizierend angesehen werden. Dennoch sind einige grundlegende Erkenntnisse und Größendimensionen ableitbar. In weiteren Forschungsarbeiten sollten Modelle zur Ressourcenanalyse entwickelt werden, die eine komplexere Abbildung der Realität erlauben.

Literatur

- [AnKu02] Anderson, D.; Kubiawicz, J. (2002): Der Weltcomputer, in: Spektrum der Wissenschaft Juni 06/2002, S. 80-87
- [BaBa96] Bamberg, G.; Baur, F.(1996): Statistik, 9. Aufl., München, 1996
- [Bark01] Barkei, D. (2001): Peer-to-Peer Computing – Technologies for Sharing and Collaborating on the Net, Hillsboro, 2002
- [Cen02] CenterSpan: A Content – Centric Distribution Strategy: The Secure, Mediated Peer-to-Peer Content Delivery Network as Best Value for Rich Downloadable Content and on-Demand Streaming, White Paper, www.centerspan.com, 2002
- [Cou+02] Coulouris, G.; Dollimore, J.; Kindberg T. : Verteilte Systeme, 3. Aufl., München, 2002
- [Des+01] Deshpande, H.; Bawa, M.; Garcia-Molina, H.: Streaming Live Media over Peers. Stanford Database Group Technical Report, 2001
- [Erpen99] Erpenbeck, A. : Streaming Media und Real Video, 1999, online: www-lehre.informatik.uni-osnabrueck.de/~aerpenbe/papers/real.pdf, abgerufen am 05.11.2002
- [GeAn02] Gehrke, N.; Anding, M.: A Peer-To-Peer Business Model for the Music Industry in: Monteiro, J.L./ Swatman, P.M./ Tavares, L.V. (Hrsg.): Towards the Knowledge Society, Proceedings of the 2nd IFIP Conference on eCommerce, eBusiness and eGovernment, Lissabon (Kluwer Academic Publishers), S. 243 - 257, 2002
- [Ge+02] Gehrke, N., Burghardt, M., Schumann, M. : Ein Peer-to-Peer basiertes Modell zur Dezentralisierung elektronischer Marktplätze in: Weinhardt, C., Holtmann, C. (Hrsg.): E-Commerce - Netze, Märkte, Technologien, S. 101-166, Heidelberg, 2002.
- [Info02] Infoquelle, http://www.infoquelle.de/InternetWorld/Streaming_Media.cfm, abgerufen am 05.11.02
- [Leu02] Leuf, B.: Peer-to-Peer – Collaboration and Sharing over the Internet, Addison Wesley, Indianapolis, 2002
- [Min01] Minar, N.: Distributed Systems Topologies: Part 1, 2001, online: http://www.openp2p.com/pub/a/p2p/2001/12/14/topologies_one.html, abgerufen am 07.08.2002
- [Min02] Minar, N.: Distributed Systems Topologies: Part 2, 2002, online: http://www.openp2p.com/pub/a/p2p/2002/01/08/p2p_topologies_pt2.html, abgerufen am 07.08.2002
- [NgZa02a] Nguyen, T.; Zakhor, A., Protocols for Distributed Video Streaming, ICIP 2002, Rochester NY, USA
- [NgZa02b] Nguyen, T.; Zakhor, A.: Distributed Video Streaming over the Internet, in Proceedings of SPIE Conference on Multimedia Computing and Networking, San Jose, California, January 2002

- [Riv+78] Rivest, R.; Shamir, A. ; Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM 21,2 (Feb. 1978), 120-126
- [Ros+02] Rosenblatt, B.; Trippe, B.; Mooney, S.: Digital Rights Management -Business and Technology. M&T Books, New York 2002
- [Schl96] Schlittgen, R.: Statistische Inferenz, München, 1996
- [Shir01] Shirky, C.: Listening to Napster. In: Oram A (ed) Peer-to-Peer, Harnessing the Power of Disruptive Technologies. O'Reilly, Sebastopol, 2001, S 21-38
- [Stein99] Steinmetz, R.: Multimedia-Technologie, 2. Aufl., Heidelberg, 1999
- [Vo00] Voß, W.; u.a.: Taschenbuch der Statistik, München, Wien, 2000
- [Wirz01] Wirtz, B. (2001): Medien- und Internetmanagement, 2. Aufl., Wiesbaden, 2001
- [Xu+B02] Xu, D.; Hefeeda, M.; Hambrusch, S.; Bhargava, B.: On Peer-to-Peer Media Streaming, Proc. of IEEE International Conference on Distributed Computing Systems (ICDCS 2002), Wien, Austria, July 2002
- [Zwi01] Zwißler, S. : Electronic Commerce – Electronic Business, Heidelberg, 2001